УДК 519.71

Х. Виссия, В.В. Краснопрошин, А.Н. Вальвачев
Белорусский государственный университет, г. Минск, Республика Беларусь
vissia@bsu.by, krasnoproshin@bsy.by, van@bsu.by

Технология выполнения IT-проектов коллективами распределенных исполнителей

В статье предлагается программная технология для реализации IT-проектов распределенными группами исполнителей. Предложены механизмы моделирования проекта, формирования предметной области и ее отображения в компьютерную систему. Представлены результат их реализации в рамках многоагентного подхода и примеры использования для решения практических задач.

Введение

В настоящее время теория управления проектами находится в стадии адаптации к формирующейся информационно-коммуникативной среде [1]. Под проектом в общем случае понимается решение некоторой задачи, включающее постановочную, техническую и описательную составляющие. Первая часть – постановка задачи, цель и определение требований к решению. Вторая – деятельность, включая планирование, выполнение, контроль уровня достижения цели. Третья – написание отчета и руководства по использованию результата проекта.

В результате глобализации «производственная» парадигма развития общества изменилась на «информационную», поэтому доминируют информационно-технологические проекты. Более того, компьютеризация привела к тому, что любой проект должен иметь отображение в форме, доступной для представления на компьютере, и включать соответствующие знания (Knowledge Base System, KBS).

Реализация конкурентоспособных проектов основана на использовании инновационных знаний (Knowledge Technology Project, (KTP)), независимо от места расположения их источников. Под инновационными знаниями будем понимать структурированную и формализованную информацию, необходимую и достаточную для реализации проекта и обладающую свойствами актуальности, достоверности и полноты.

В условиях ускорения динамики бизнес-процессов знания быстро устаревают, поэтому одной из актуальных проблем является разработка технологий для оперативной реализации KTP.

Разработка KTP в общем случае включает четыре типовых процесса [1], [2]:

1) построение модели проекта и его декомпозицию на подзадачи;
2) добычу знаний, релевантных подзадачам;
3) интеграцию знаний отдельных подзадач в единую предметную область;
4) создание программы для отображения проекта в компьютерную систему.

Для их выполнения разработаны технологии, основными из которых являются Lotus Notes/Domino и Outlook/Projects/Exchange. Они эффективны в больших компаниях при реализации крупных проектов [3]. Другие технологии, в частности, eGroupWare, Tutos, PHProjekt требуют существенного участия программиста для преобразования фрагментов в предметную область. Попытки исключить программирование с помощью технологий Rational Rose, Rational XDE, Rational Requisite Pro, Rational Rapid Developer, Rational ClearCase не всегда дают результат, так как автоматически генерируемый код часто требуется корректировать вручную высококвалифицированному программисту [4].

Таким образом, можно сделать вывод, что в настоящее время актуальна проблема разработки технологий реализации КТР, исключающих или минимизирующих роль инженера знаний и программиста. Кроме того, эти технологии должны поддерживать выполнение проектов территориально распределенными группами исполнителей, обеспечивать независимость времени реализации КТР от увеличения количества исполнителей, автоматическое отображение материалов проекта в компьютерную систему [5].

В данной статье представлены результаты разработки интуитивно понятных и не требующих программирования механизмов реализации КТР, решающие перечисленные задачи.

Основные определения, модели, аксиомы

Для формирования цельного взгляда на проблему дадим определения базовым понятиям и построим соответствующие модели. Для этого будем использовать онтологический подход, стандартизированный в IDEF5 и допускающий уточнение каждого элемента первичной грубой модели до уровня исполнительного программного модуля. Представленные ниже определения не противоречат уже существующим в информатике, а лишь уточняют их с точки зрения конструктивности в рамках данного класса задач.

Глобальная информационно-коммуникативная среда (ГИКС) – инфраструктура, в рамках которой осуществляется проект. Элементами ГИКС является Интернет, корпоративные сети и т.д.

Проект – совокупность четырех составляющих: постановки задачи (S), требований к решению *(Req)*, результата выполнения проекта (Decision) и руководства пользователя *(Guide)* по использованию результата:

$$Project = (S, Req, Decision, Guide).$$

Группа – коллектив исполнителей проекта, включающего центр (Center), инициирующий проект, и удаленных исполнителей (E), реализующих проект:

$$Group = (Center, E^1, E^2, E^n).$$

Сцена – схема, включающая участников выполнения проекта, их роли отношения и средства информационного обмена (Com) в рамках ГИКС:

$$Scene = (Project, Group, Com).$$

Свойство структурированности любого проекта требует декомпозиции S на подзадачи S^1, S^2, S^n. Оперативность выполнения КТР говорит о критичности затрат времени (T) и средств (M) на выполнение проекта. Соответственно уточним модель проекта:

$$Project = (S, S^1, S^2, S^n, Req^1,...,Req^k, Decision^1,..., Decision^n, Guide^1,...,Guide^n, T^1,...,T^n, M^1,...,M^n),$$

где: $n \rightarrow \infty$, $k = variable$.

Знания – структурированная и формализованная информация *(Decision)*, необходимая для реализации проекта. В данном случае знания эквивалентны решениям подзадач проекта:

$$Decision = \sum Decision^i.$$

Паттерн знаний – знания, необходимые и достаточные для решения подзадачи, включающие четыре составляющих: постановку подзадачи, теоретическое решение *(dTheor*), практическое решение (dPrac) и соответствующе руководство пользователя:

$$Pat^i = Decision^i = (S^i, dTheor^i, dPrac^i , Guide^i).$$

Посредник – мобильная программа, обеспечивающая перемещение паттерна между центром и исполнителями в рамках ГИКС:

$$Mediator = (adrCenter,\ adrE^i,\ Pat^i),$$

где: *adrCenter* – адрес центра, $AdrE^i$ – адрес i-го исполнителя.

Предметная область проекта (ПрО) – совокупность паттернов, необходимых и достаточных для реализации проекта:

$$subPat = \sum Pat^i.$$

Результат выполнения проекта – отображение предметной области subPat в KBS, включающую ПрО, средство навигации *(Menu)*, средство отображения элементов ПрО (контента) на экран *(Content)*, программы стандартного системного сервиса *(Service)* и специализированной обработки *(Proc)*:

$$Sys = (PrO,\ Menu = f'(PrO),\ Content = f''(PrO,\ Menu),\ Service,\ Proc).$$

На базе определений введем аксиомы, отражающие эволюцию S в KBS.

Аксиома 1. Центр и исполнители позиционированы в ГИКС.

Аксиома 2. Обмен информацией между центром и исполнителями осуществляет коммуникатор, ориентированный на использование инфраструктуры Интернет.

Аксиома 3. Предметная область считается цельной, если она содержит паттерны знаний, обеспечивающих решение всех подзадач проекта.

Аксиома 4. Проект считается выполненным, если предметная область отображена в компьютерную систему, обеспечивающую реализацию проекта.

На основе определений и аксиом сформулируем общую постановку задачи.

Постановка задачи

Пусть имеется проект *Project* и группа для его реализации *Group*. Центр формулирует задачу *S*, декомпозирует ее на подзадачи S^1, S^2, S^n, исполнители E^1, E^2, E^n их решают в соответствии с требованиями $Req^1,\ldots,Req^k$. Совокупность решений $Decision^1,\ldots,\ Decision^n$, представленная в форме *Sys*, является результатом выполнения проекта. Требуется разработать технологию преобразования *S* в *Sys*, обеспечивающую стабилизацию времени реализации проекта при увеличении количества исполнителей, то есть $T(n) \approx T(m)$ при $n = const,\ m \rightarrow \infty$.

Решение общей задачи в рамках указанных выше процессов 1 – 4 требует решения следующих подзадач:

– построение модели *S* и декомпозиция ее на подзадачи $S^1,\ S^2,\ S^n$, отправка фрагментов необогащенной модели Pat^i исполнителям;

– обогащение Pat^i фрагментов исполнителями за счет включения решений для каждого фрагмента проекта;

– интеграция фрагментов Pat^i в ПрО;

– отображение ПрО в *Sys*.

Основные требования к решению: 1) использование стандартных IBM-совместимых персональных компьютеров; 2) использование OS Windows XP; 3) интуитивно понятный интерфейс.

Модель предметной области

В основе решения поставленных задач лежит предметная область, следовательно, прежде всего необходимо разработать ее модель. В настоящее время для этого обычно используется подход Питера Чена. Он структурирует семантическую информацию о ПрО,

выделяя значимые сущности (entity) и связи (relationship) между ними в форме иерархии, известной как ERD-модель [3], [4]. Применить ее для распределенных ПрО достаточно сложно, так как она изначально ориентирована на локальные группы исполнителей.

Для решения этой проблемы предлагается динамическая модель ПрО *mPat*, состоящая из двух частей (рис. 1). Статическая часть представляет собой граф, вершиной которого является задача *S*, узлы определяют иерархию подзадач $S^1,..,S^n$, дуги – уровень их вложенности. Динамическая часть представлена терминальными вершинами, включающими решения Pat.

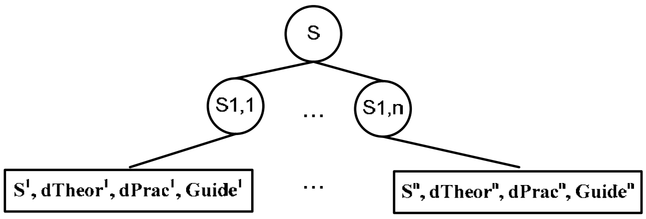


Рисунок 1 – Динамическая модель ПрО

Модель проекта, где $dTheor^i = \varnothing$, $dPrac^i = \varnothing$, $Guide^i = \varnothing$, назовем необогащенной. Модель проекта, где $dTheor^i \neq \varnothing$, $dPrac^i \neq \varnothing$, $Guide^i \neq \varnothing$, назовем обогащенной.

Таким образом, реализация проекта состоит из построения модели *mPat*, ее обогащения и отображения в компьютерную систему. Для этого требуется сформировать типовую сцену решения, определив ее участников, их роли, отношения и коммуникации.

Сцена решения и процессы ее реализации

В качестве основы для построения технологии предлагается организационно-коммуникативная схема реализации проектов, отражающая типовые этапы разработки KBS (рис. 2). Она построена на основе теории активных систем [6], модифицированной за счет использования приведенных выше моделей центра, исполнителей и посредника.

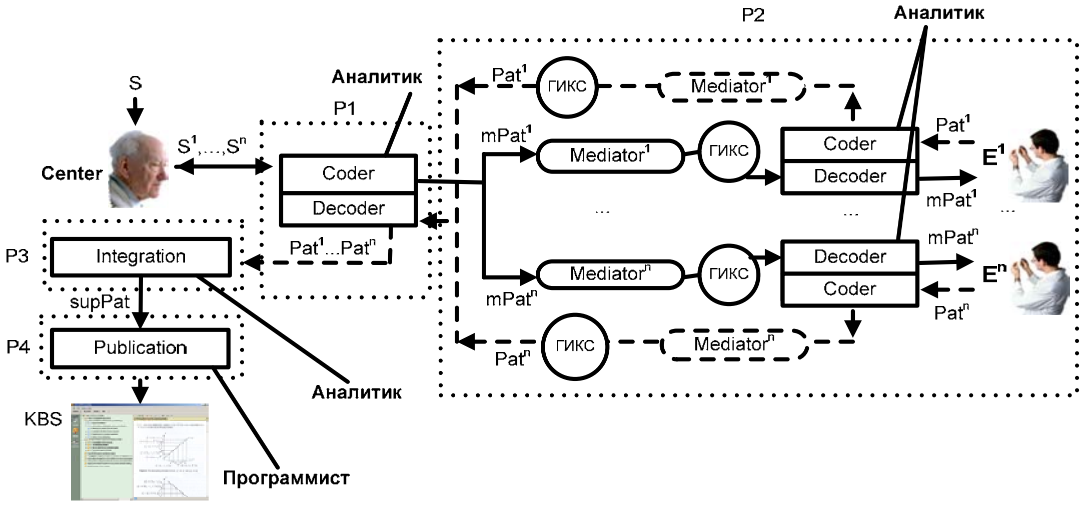


Рисунок 2 – Организационно-коммуникативная схема выполнения IT-проектов

Согласно сцене, в создании целевого продукта участвуют Центр, исполнители, аналитик (инженер знаний) и программист. Роль аналитика и программиста в проекте велика, поэтому ее сокращение является существенным вкладом в ускорение разработки KBS. Для этого необходимо автоматизировать процессы P1 – P4. Для этого прежде всего выделим подпроцессы.

Процесс P1 (моделирование) включает три подпроцесса: 1) построение в центре модели предметной области (ПрО) проекта; 2) декомпозиция модели на «необогащенные» паттерны; 3) доставка «необогащенных» паттернов экспертам.

Процесс P2 (обогащение) включает два подпроцесса: 1) включение в паттерн требуемых экспертных знаний; 2) отправка паттерна в центр.

Процесс P3 (интеграция) включает три подпроцесса: 1) анализ полноты количества паттернов; 2) нормализация информации; 3) интеграция паттернов в ПрО проекта.

Процесс P4 (публикация) включает пять подпроцессов: 1) анализ ПрО; 2) выделение контента; 3) построение дерева управления контентом; 4) формирование меню и базы данных целевой системы из дерева управления и контента; 5) сборка целевой системы.

Сцена носит распределенный характер, поэтому для реализации процессов P1 – P4 целесообразно применить многоагентный подход, изначально ориентированный на работу в Интернет [7], [8].

Архитектура, реализация, применение системы

На основе приведенных выше моделей для процессов P1 – P4 разработаны четыре программных агента, которые обозначим как Modeller, Miner, Integrator, Publicator. В результате их адаптации в сцену (рис. 2) получим многоагентную архитектуру (рис. 3).

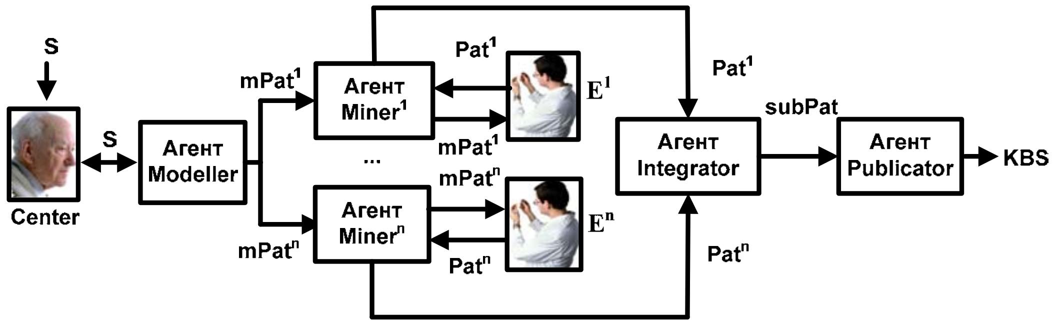


Рисунок 3 – Многоагентная архитектура

Данная архитектура реализована на языке C# в рамках платформы .Net. Использован многослойный подход: агенты Modeller, Integrator, Publicator объединены в программе SDP Workshop для центра (рис. 5а). Агент Miner интегрирован с посредником Mediator в программе SDP Expert для исполнителей (рис. 4).

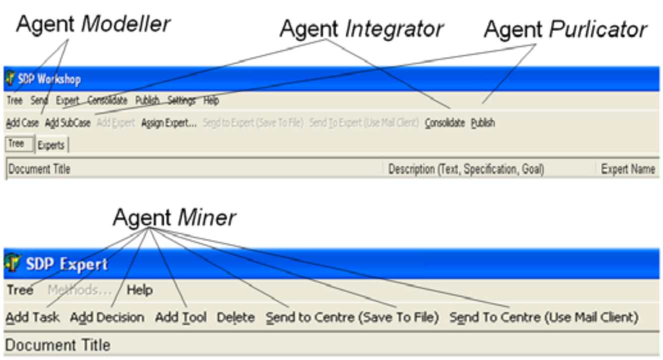


Рисунок4 – Интерфейсы центра и исполнителя: WorkShop (4a), SDP Expert (4b)

Возможны два варианта отображения ПрО: сетевой и локальный. В первом случае ПрО размещается с помощью стандартных средств на сервере и просматривается пользователями посредством браузеров MS Explorer, Opera, FireFox (рис. 5).

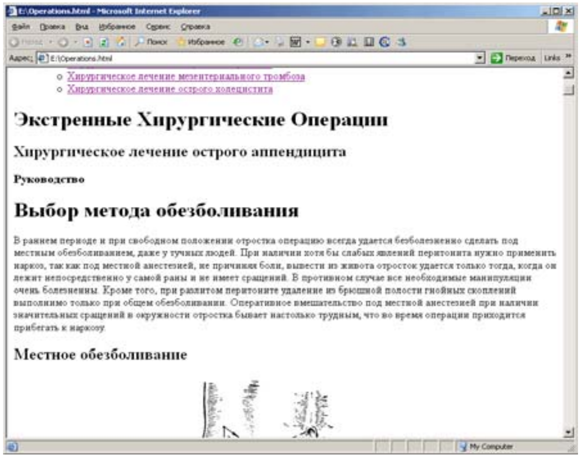


Рисунок 5 – Сетевой вариант интерфейса

Во втором случае формируется система на CD с возможностью использования на компьютере, не подключенном к Интернет (рис. 6). Локальная KBS может иметь один из четырех возможных вариантов интерфейса.

Предложенная многослойная архитектура использовалась для реализации web-проекта «Программирование на языке Delphi», локальных проектов «Экстренные хирургические операции», «Orthopedic cases», «Atlas Forensic Medicine».

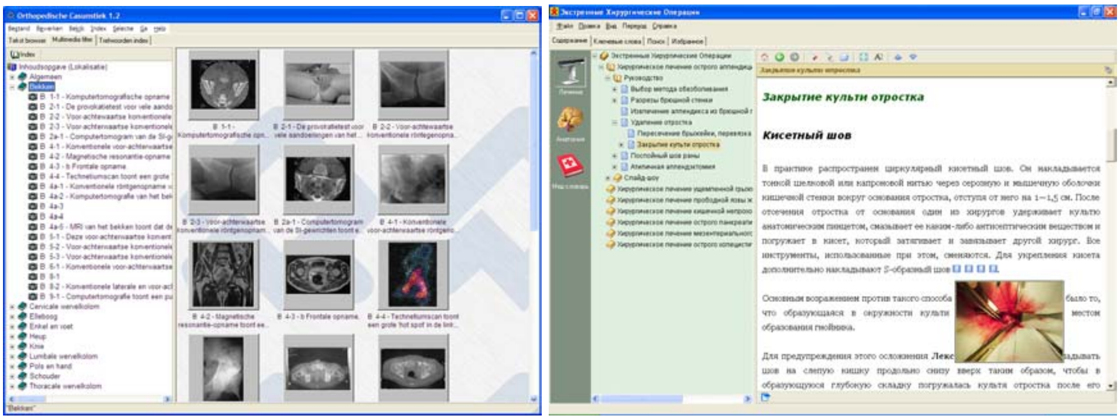


Рисунок 6 – Локальные варианты интерфейса

В данных проектах за счет исключения роли программиста удалось снять проблему наличия ошибок времени выполнения, столь характерную для больших систем. Аналитик в них также не был задействован, так как модель и декомпозиция проектов выполнялась непосредственно заказчиками. В целом эффективность достигается значительным сокращением времени и средств на выполнение проекта.

Заключение

В статье рассмотрены проблемы построения технологии для оперативного выполнения IT-проектов территориально распределенными исполнителями.

Предложен комплекс определений, уточняющий с конструктивной точки зрения известные понятия ИИ в рамках тематики IT-проектов. Построена общая сцена выполнения проектов, выделены типовые процессы ее реализации. Для каждого процесса разработан комплекс моделей, описывающий полный жизненный цикл выполнения

IT-проектов в инфраструктуре Интернет. На основе многоагентного подхода модели реализованы в форме агентов, реализующих в рамках комплекса моделей интеллектуальные функции аналитика и программиста. Агенты реализованы на языке C# для .Net в рамках архитектуры и соответствующей технологии для автоматизации наиболее трудоемких интеллектуальных операций выполнения IT-проектов. Представлены примеры использования технологии для решения практических задач из области медицины.

Простота моделирования предметной области, реализация интеллектуальных функций аналитика и программиста, параллельные процессы добычи фрагментов знаний, автоматизация их интеграции и публикации позволили существенно сократить время выполнения IT-проектов распределенными исполнителями. Интеллектуальная деятельность аналитика и программиста реализована за счет учета в моделях элементов, специфических для данного класса задач и введения соответствующих стандартов представления знаний. Модуль обработки предметной области стандартизировать не представляется возможным, т.к. он специфичен для каждого проекта.

Развитие технологии предполагает увеличение количества возможных вариантов интерфейса KBS и библиотек обработки, включаемых в целевую систему в зависимости от семантики предметных областей.

Литература

1. Васильев Д.К. Типовые решения в управлении проектами. – М.: ИПУ РАН, 2003. – 75 с.
2. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. – СПб.: Питер, 2002. – 496 с.
3. Элиот М. Lotus Notes Developer's Toolbox. Советы по быстрому и успешному внедрению приложений. – М.: Кудиц-Пресс, 2007. – 688 с.
4. Боггс У., Боггс М. UML и Rational Rose. – М.: Лори, 2000. – 580 с.
5. Краснопрошин В.В., Шаках Г., Вальвачев А.Н. Интеграция распределенных экспертных знаний: проблемы и решения // Информатика. – 2004. – № 1. – С. 45-53.
6. Новиков Д.А., Петраков С.Н. Курс теории активных систем. – М.: СИНТЕГ, 1999. – 108 с.
7. Krasnoproshin V.V. Knowledge–Based Systems: Problems and Solutions // Proceedings of the International Conference on Modeling and Simulation (MS'2006). – Konya (Turkey). – Vol. 1. – P. 1-4.
8. Wooldridge M. Multiagent Systems. – John Wiley & Sons, 2002. – 340 p.

Х. Віссія, В.В. Краснопрошин, А.М. Вальвачев

Технологія виконання IT-проектів колективами розподілених виконавців

У статті пропонується програмна технологія для реалізації IT-проектів розподіленими групами виконавців. Запропоновані механізми моделювання проекту, формування наочної області і її відображення в комп'ютерну систему. Представлені результат їх реалізації в рамках багатоагентного підходу і приклади використання для розв'язання практичних задач.

H. Vissia, V.V. Krasnoproshin, A.N. Valvachev

Technology for Realizing IT-Projects by Distributed Developers

The paper describes a technology for realizing IT-projects by groups of distributed developers. Mechanisms for project modeling, knowledge domain formation and its representation into a computer system are proposed. Results and examples of practical implementation within a multi-agent approach are presented.

Статья поступила в редакцию 18.07.2008.